

CLAIMS

At the time of the Action:

Pending Claims: 1-13 and 16-40

Withdrawn Claims: None

Canceled Claims: 14-15

After this Response:

Pending Claims: 1-13, 16-27 and 35

Amended Claims: 27 and 35

Withdrawn: None

Canceled Claims: 14-15, 28-34 and 36-40

New Claims: None

1. **(Previously Presented)** A computer implemented system, comprising:

a computing system having a decentralized operating system unifying and orchestrating one or more services and resources represented as services executing on the computing system, the decentralized operating system comprising:

a process kernel, and

a distributing kernel;

wherein the process kernel represents the one or more services as processes, manages the processes, and facilitates communication between the processes-, each of the one or more services coupled to the decentralized operating system comprises an autonomous computation entity that exchanges one or more messages with each of the

one or more services coupled to a disparate decentralized operating system that resides in a different trust domain with a different security policy based in part on a protocol specified by each of the one or more services coupled to the disparate decentralized operating system;

a port associated with each of the one or more services that is endued with one or more behavioral types that are specified by a unilateral contract, communication between the one or more services representing disparate resources is established based in part on compatibility between the one or more behavioral types associated with each of the one or more services;

wherein the distributing kernel distributes computation for each of the one or more services coupled to the distributing kernel so as to control and coordinate resources, such that the one or more services representing the resource perform computations on a plurality of computers linked by a communication network; and

wherein further a memory coupled to a processor retains the decentralized operating system.

2. (Previously Presented) The computer system of Claim 1, wherein the computer system further comprises:

a microcomputer,

a personal digital assistant,

a cellular phone, or

a display.

3. **(Previously Presented)** The computer system of Claim 1, wherein the one or more services comprises:

a communication primitive;

a behavioral primitive that comprises a unilateral contract; and

a designation primitive that comprises a port identifiable by an identifier that includes a uniform resource identifier.

4. **(Previously Presented)** The computer system of Claim 3, wherein the port is endued with a behavior type as specified by the unilateral contract.

5. **(Previously Presented)** The computer system of Claim 3, wherein the unilateral contract defines a protocol for exchanging messages in a particular order with a service to which the unilateral contract belongs.

6. **(Previously Presented)** The computer system of Claim 5, wherein the communication primitive comprises a set of message types usable in the messages exchanged among services so as to call a service to perform a certain task.

7. **(Previously Presented)** The computer system of Claim 6, wherein the decentralized operating system further separates the control information from the data information in the messages when the messages are exchanged.

8. **(Previously Presented)** The computer system of Claim 1, wherein the autonomous computation entities comprise services representing one or more of: devices, content, applications, or people.

9. **(Previously Presented)** A computer implemented networked system for networking computer systems, comprising:

 a first decentralized operating system executing on a computer system, which comprises:

 a first process kernel representing a first set of resources as a first set of services;

 a first distributing kernel designating uniform resource identifiers for the first set of services and distributing messages among the first set of services, each of the first set of services including a unilateral contract, the unilateral contract expressing behaviors of each of the first set of services;

 a second decentralized operating system executing on a disparate computer system coupled to the networked system, which comprises:

 a second process kernel representing a second set of resources as a second set of services;

 a second distributing kernel designating uniform resource identifiers for the second set of services distributing messages among the second set of services, each of the second set of services including a unilateral contract, the unilateral contract expressing behaviors of each of the second set of services;

 wherein communication between the first and second set of services is

established based in part on compatibility determined between behavioral types specified by the unilateral contract associated with each of the first and second set of services;

wherein further at least one of the resources from the second set of resources being represented as services from the second set of services is unified to and orchestrated by the first process kernel of the first decentralized operating system, the services performing computations on a plurality of computers linked by communication network; and

wherein further a memory coupled to a processor retains at least one of the first or the second decentralized operating system.

10. **(Previously Presented)** The networked system of Claim 9, wherein the first and second set of services comprise device drivers for devices.

11. **(Previously Presented)** The networked system of Claim 9, further comprising the process kernel communicating messages as processes among the first and second set of services.

12. **(Previously Presented)** The networked system of Claim 11, further comprising an operating system kernel that manages memory, controlling devices, maintaining time and date, and allocating system resources.

13. **(Original)** The networked system of Claim 9, further comprising a network coupled to the first computer system, the network is selected from a group consisting of high bandwidth, low latency systems; high bandwidth, high latency systems; low bandwidth, high latency systems; and low bandwidth, low latency systems.

14. **(Canceled)**

15. **(Canceled)**

16. **(Previously Presented)** The networked system of Claim 13, wherein a service from the second set of services registers with the first distributing kernel to obtain a uniform resource identifier.

17. **(Previously Presented)** The networked system of Claim 13, wherein the first distributing kernel distributes a message to a service from a first set of service, the message being sent by a service from a second set of services.

18. **(Previously Presented)** The networked system of Claim 13, wherein the first decentralized operating system orchestrates a composition of a service from a first set of services and a service from a second set of services.

19. **(Previously Presented)** A computer implemented system,

comprising:

a decentralized operating system unifying and orchestrating services and resources that comprises:

a process kernel, and

a distributing kernel,

wherein the process kernel represents the resources as one or more services, wherein a particular service is an autonomous computation entity that exchanges one or more messages with a disparate service that resides in a different trust domain with a different security policy based in part on a protocol specified by the particular service; and

wherein the distributing kernel comprises;

a URI (Uniform Resource Identifier) manager that manages names, each of the names constituting a unique designation of the one or more services at the computer system so that the one or more services can be discovered, the one or more services perform computations on a plurality of computers linked by communication network; and

a message dispatcher that forwards messages among the one or more services, each of the one or more services being identifiable by a name managed by the URI manager, each of the one or more services comprises a port that is endued with one or more behavioral types that are specified by a unilateral contract, communication between the one or more services is established based in part on

compatibility between the one or more behavioral types associated with each of the one or more services;
wherein a memory coupled to a processor stores at least one of the URI manage or the message dispatcher.

20. **(Previously Presented)** The computer system of Claim 19, wherein the distributing kernel further comprises a security manager that controls authentication and authorization of rights and restrictions among the one or more services.

21. **(Previously Presented)** The computer system of Claim 19, wherein the distributing kernel further comprises a service loader that executes a sequence of instructions for loading components and the one or more services, the service loader being capable of dynamically loading or unloading the one or more services during the operation of the decentralized operating system.

22. **(Previously Presented)** The computer system of Claim 19, wherein the URI manager receives a register message from a particular service from the one or more services to obtain a unique designation, assigns the unique designation to the particular service and further receives an unregister message for removing an assigned unique designation from a registry.

23. **(Original)** The computer system of Claim 19, wherein the message

dispatcher forwards a message from a first service to a second service if the first service has a first uniform resource identifier assigned by the URI manager and the second service has a second uniform resource identifier assigned by the URI manager.

24. **(Previously Presented)** The computer system of Claim 19, wherein the message dispatcher comprises a message validity verifier that verifies that a message sent to the message dispatcher is a message whose structure complies with the SOAP protocol.

25. **(Previously Presented)** The computer system of Claim 19, further comprising a network manager that distributes messages forwarded by the message dispatcher to another computer system.

26. **(Original)** The computer system of Claim 25, wherein the network manager comprises a serializer/deserializer, a transmission protocol processor, and a control/data plane separator.

27. **(Currently Amended)** A method implemented on a computer system, comprising:

initializing a decentralized operating system on one or more computing systems, the decentralized operating system unifying and orchestrating services and resources represented as services executing on the one or more computing systems, the decentralized operating system further comprising a process kernel and a distributing kernel;

assigning a first unique name to a first service upon request, the first service represented as a process by the process kernel of a first computing system, the first service including a first unilateral contract for expressing the behaviors of the first service; and

distributing a message by the distributing kernel to the first service using the first unique name, the message being sent by a second service having a second unique name, the second service including a second unilateral contract for expressing the behaviors of the second service, the second service resides in a different trust domain with a different security policy compared to the first service, communication between the first service and the second service via the message is established when the first unilateral contract is compatible with the second unilateral contract,

wherein the process kernel represents the first and second services as a respective resource[.];

loading a network manager and other services according to instructions written in a customizable, tag-based language;

spawning a service to listen for incoming messages for the first service that has been assigned the first unique name;

rejecting the message without distributing the message when a structure of the message fails to comply with a protocol for exchanging structured and type information of messages written in a customizable, tag-based language;

forwarding the message to the first service without routing the message through the network manager when the first service and the second service runs on a common computer system; and

forwarding the message by transmitting data information separately from control
information to the first service by routing the message through the network manager when the
first service runs on the first computer system whereas the second service runs on a second
computer system.

28. - 34. (Canceled)

35. (Currently Amended) A computer-readable medium storing computer-
executable instructions that, when executed, cause one or more processors to perform
~~thereon for implementing a method, the a method comprising:~~

initializing a decentralized operating system on one or more computing systems, the decentralized operating system unifying and orchestrating services and resources represented as services executing on the one or more computing systems, the decentralized operating system further comprising a process kernel and a distributing kernel;

assigning a first unique name to a first service upon request, the first service represented as a process by the process kernel of a first computing system, the first service including a first unilateral contract for expressing the behaviors of the first service; and distributing a message by the distributing kernel to the first service using the unique name, the message being sent by a second service having a second unique name, the second service including a second unilateral contract for expressing the behaviors of the second service, the message establishes communication between the first service and second service when the first unilateral contract is compatible with the second unilateral contract, the second service resides

in a different trust domain with a different security policy compared to the first service, a resource is represented by multiple services[.]];

loading a network manager and other services according to instructions written in a customizable, tag-based language;

spawning a service to listen for incoming messages for the first service;

rejecting the message when a structure of the message fails to comply with a protocol written in the customizable, tag-based language;

forwarding the message to the first service without routing the message through the network manager when the first service and the second service runs on a common computer system; and

forwarding the message by transmitting data information separately from control information to the first service through the network manager when the first service runs on the first computer system whereas the second service runs on a second computer system.

36. – 40. (Canceled)